



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

komputer adalah sebuah mesin atau alat bantu elektronik yang secara otomatis mampu mengolah data perhitungan serta melakukan proses *input*, *output* untuk memproses dan menerima informasi.

Komputer adalah sebuah mesin yang dapat dikendalikan melalui perintah (*programmable machine*) yang dirancang untuk secara otomatis melakukan serangkaian urutan penghitungan (*arithmetic*) atau proses-proses yang diurutkan secara logis (Hartono, 2013: 27).

Komputer berasal dari kata latin *computare* yang berarti menghitung (Pratama, 2014: 14).

Komputer adalah alat bantu pemrosesan data secara elektronik terdiri dari *input*, *output* yang saling bekerja secara otomatis untuk menerima data, memprosesnya dan menerima informasi (Asropudin, 2013: 19).

2.1.2 Pengertian Perangkat Lunak

Perangkat lunak adalah serangkaian instruksi dengan aturan tertentu yang mengatur operasi perangkat keras.

Perangkat lunak (*software*) adalah kumpulan beberapa perintah yang dieksekusi oleh mesin komputer dalam menjalankan pekerjaannya (Sujatmiko, 2012: 256).

Perangkat Lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*) (Rosa dan Shalahuddin, 2013: 2).

Program komputer, yang dikenal dengan perangkat lunak, adalah instruksi-instruksi untuk komputer, yang mendeskripsikan apa yang harus dilakukan komputer (Sianipar, 2013: 8).



2.1.3 Pengertian Internet

Internet adalah interkoneksi jaringan komputer global atau skala besar yang menyediakan berbagai informasi dan fasilitas informasi yang dihubungkan menggunakan protokol khusus.

Internet adalah interkoneksi jaringan komputer skala besar (mirip *Wide Area Network*), yang dihubungkan menggunakan protokol khusus (Sofana, 2013: 5).

Internet adalah suatu sistem jaringan komunikasi beberapa komputer yang terhubung tanpa batas waktu maupun tempat, sehingga dapat dikatakan sebagai suatu komunitas jaringan global (Yugianto dan Rachman, 2012: 36).

Internet (*interconnected network*) adalah jaringan komputer yang menghubungkan antar jaringan secara global, internet dapat juga disebut jaringan dalam suatu jaringan yang luas (Sibero, 2014: 10).

2.1.4 Pengertian Aplikasi

Aplikasi adalah perangkat lunak komputer yang digunakan untuk perintah program yang dibuat untuk mengerjakan pekerjaan khusus.

Istilah Aplikasi berasal dari bahasa Inggris, yaitu *application* yang berarti program komputer yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word* dan *Microsoft Excel* (Sujatmiko, 2012: 22).

Application atau aplikasi adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word* dan *Microsoft Excel* (Asropudin, 2013: 6).

Aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya (Sutabri, 2012: 147).

2.1.5 Pengertian Sistem Pendukung Keputusan

Kusrini (15:2007), sistem pendukung keputusan merupakan system informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data yang digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat.



2.1.6 Pengertian Pegawai

Pegawai adalah setiap orang pribadi, yang melakukan pekerjaan berdasarkan perjanjian atau kesepakatan kerja, baik tertulis, termasuk yang melakukan pekerjaan dalam jabatan negeri atau badan usaha milik Negara atau badan usaha milik daerah (Muljono, 2010:113).

2.1.7 Pengertian *Tichnique for Order Performance by Similarity to Ideal Solution (TOPSIS)*

Nofriansyah (27 : 2014), Topsis adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang pada tahun 1981. Topsis menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal positif dan jarak terpanjang (terjauh) dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak Euclidean (jarak antara dua titik) untuk menentukan kedekatan relatif dari suatu alternatif.

Metode Topsis didasarkan pada konsep bahwa alternatif terpilih yang terbaik tidak hanya memiliki jarak terpendek dari solusi ideal positif tetapi juga memiliki jarak terpanjang dari solusi ideal negatif. Tahapan metode Topsis :

1. Membuat matriks keputusan yang ternormalisasi.
2. Membuat matriks keputusan yang ternormalisasi terbobot.
3. Menentukan matriks solusi ideal positif dan matriks solusi ideal negative.
4. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matriks solusi ideal negative.
5. Menentukan nilai preferensi untuk setiap alternative

Adapun langkah-langkah algoritma dari metode Topsis adalah :

1. Menentukan normalisasi matriks keputusan. Nilai ternormalisasi rij dihitung

dengan rumus
$$rij = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}$$

Keterangan :

$i = 1, 2, \dots, m$; dan

$j = 1, 2, \dots, n$,



2. Menentukan bobot ternormalisasi matriks keputusan. Nilai bobot ternormalisasi y_{ij} sebagai berikut :

$$y_{ij} = w_{ij}r_{ij};$$

Keterangan :

$i = 1, 2, \dots, m$

$J = 1, 2, \dots, n.$

$$A^+ = (y_1^+, y_2^+, \dots, y_n^+);$$

$$A^- = (y_1^-, y_2^-, \dots, y_n^-);$$

Dengan

$$y_j^+ = \begin{cases} \max y_{ij}; & \text{jika } j \text{ adalah atribut keuntungan} \\ \min y_{ij}; & \text{jika } j \text{ adalah atribut biaya} \end{cases}$$

$$y_i^- = \begin{cases} \max y_{ij}; & \text{jika } j \text{ adalah atribut keuntungan} \\ \min y_{ij}; & \text{jika } j \text{ adalah atribut biaya} \end{cases}$$

Dengan nilai $j = 1, 2, \dots, n.$

3. Jarak antara alternatif A_i dengan solusi ideal positif dirumuskan :

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_i^+ - y_{ij})^2};$$

Keterangan :

$i = 1, 2, \dots, m.$

4. Jarak antara alternatif A_i dengan solusi ideal negatif dirumuskan sebagai :

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij} - y_i^-)^2};$$

Dengan $i = 1, 2, \dots, m.$

5. Nilai prefensi untuk setiap alternatif (V_i) diberikan sebagai :

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}; i = 1, 2, \dots, m.$$



2.1.8 Pengertian Aplikasi Sistem Pendukung Keputusan Untuk Menentukan Pegawai Terbaik Menggunakan Metode TOPSIS Pada RSUD Rupit Kab. Muratara

Aplikasi Sistem Pendukung Keputusan Untuk Menentukan Pegawai Terbaik Menggunakan Metode TOPSIS Pada RSUD Rupit Kab. Muratara adalah perangkat lunak komputer yang digunakan untuk menentukan pegawai terbaik pada Rumah Sakit Daerah Rupit dengan melihat hasil proses seleksi nilai kriteria yang tertinggi.

2.2 Teori Khusus

2.2.1 Kamus Data

Rosa dan Shalahuddin (2013:73), Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan). Kamus data biasanya berisi:

1. Nama-nama dari data.
2. Digunakan pada – merupakan proses-proses yang terkait data.
3. Deskripsi – merupakan deskripsi data.
4. Informasi tambahan–seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data.

Tabel 2.1 Simbol–simbol dalam kamus data

| No. | Simbol | Arti |
|-----|--------|-------------------------------------|
| 1. | = | Disusun atau terdiri atas |
| 2. | + | Dan |
| 4. | [] | Baik ...atau... |
| 5. | { }n | n kali diulang atau bernilai banyak |
| 6. | () | Data Optional |

**Lanjutan Tabel 2.1** Simbol–simbol dalam kamus data

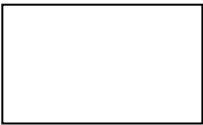
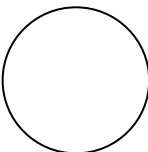
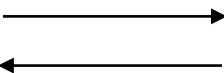

| | | |
|----|-------|--------------------------------|
| 7. | | Pemisah alternative simbol [] |
| 8. | *...* | Batas komentar |

Sumber : Rosa dan Shalahuddin (2013:72)

2.2.2 Pengertian Data Flow Diagram (DFD)

Sutabri (2012:116), *data flow diagram* adalah suatu *network* yang menggambarkan suatu sistem komputerisasi, manualisasi, atau gabungan dari keduanya, yang penggambarannya disusun dalam bentuk kumpulan komponen sistem yang saling berhubungan sesuai dengan aturan mainnya.

Tabel 2.2 Simbol DFD

| No. | Simbol | Keterangan |
|-----|---|---|
| 1. |  | <i>EXTERNAL ENTITY</i> Simbol ini digunakan untuk menggambarkan asal atau tujuan data. |
| 2. |  | PROSES Simbol ini digunakan untuk proses pengolahan atau transformasi data. |
| 3. |  | <i>DATA FLOW</i> Simbol ini digunakan untuk menggambarkan aliran data yang berjalan |
| 4. |  | <i>DATA STORE</i> Simbol ini digunakan untuk menggambarkan data <i>flow</i> yang sudah disimpan atau diarsipkan. |

Sumber: Sutabri (2012:117)



Rosa dan Shalahuddin (2013:72) menjelaskan tentang tahapan-tahapan perancangan dengan menggunakan *Data FlowDiagram* yaitu:

1. Membuat DFD Level 0 atau sering disebut juga *Context Diagram*

DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD Level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah dibuat.

3. Membuat DFD Level 2

Modul-modul pada DFD Level 1 dapat di-*breakdown* menjadi DFD Level 2. Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu di-*breakdown* lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di-*breakdown*.

4. Membuat DFD Level 3 dan seterusnya

DFD Level 3,4,5, dan seterusnya merupakan *breakdown* dari modul pada DFD Level di-atasnya. *Breakdown* pada level 3,4,5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau 2.

2.2.3 Entity Relationship Diagram (ERD)

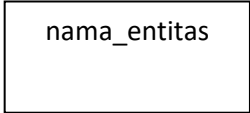
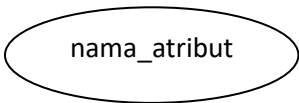
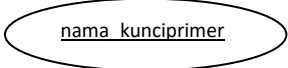
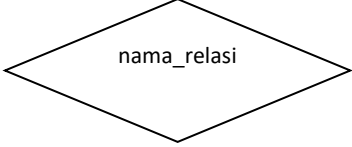
Rosa dan Shalahudin (2013:50), *entity relationship diagram* digunakan untuk pemodelan basis data relasional.

Ladjamudin (2013:142), *entity relationship diagram (ERD)* adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak.




Berikut adalah simbol-simbol yang digunakan pada *ERD* dengan notasi Chen:

Tabel 2.3 Simbol-simbol *Entity Relationship Diagram* (ERD)

| No | Simbol | Keterangan |
|----|---|--|
| 1. | Entitas atau <i>entity</i>  | Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel. |
| 2. | Atribut  | <i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas |
| 3. | Atribut kunci primer  | <i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama) |
| 4. | Relasi  | Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja |

**Lanjutan Tabel 2.3** Simbol-simbol *Entity Relationship Diagram* (ERD)

| No | Simbol | Keterangan |
|----|---|--|
| 5. | Asosiasi atau <i>association</i>  | Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B |

Sumber: Rosa dan Shalahuddin (2013:72).

2.2.4 Pengertian *Flowchart*

Ladjamudin (2013:263), *flowchart* merupakan bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah.

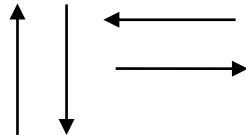
Flowchart disusun dengan simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses didalam program.

Simbol-simbol yang digunakan dapat dibagi menjadi 3 kelompok, yaitu:


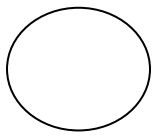
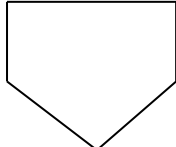
1. *Flow Direction Symbols* (Simbol penghubung atau alur)

Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol-simbol tersebut adalah sebagai berikut:

Tabel 2.4 Simbol *Flow Direction Symbols*

| NO. | SIMBOL | KETERANGAN |
|-----|---|---|
| 1. |  | Simbol arus atau <i>flow</i> Untuk menyatakan jalannya arus suatu proses |

**Lanjutan Tabel 2.4** Simbol *Flow Direction Symbols*

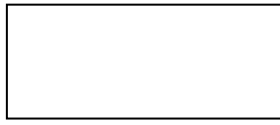
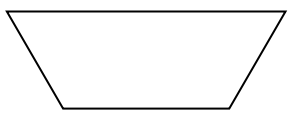
| | | |
|----|--|--|
| 2. |  | <p>Simbol Communication link</p> <p>Untuk menyatakan bahwa adanya transisi suatu daya atau informasi dari satu lokasi ke lokasi lainnya</p> |
| 3. |  | <p>Simbol Connector</p> <p>Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman atau lembar yang sama</p> |
| 4. |  | <p>Simbol Offline Connector</p> <p>Untuk menyatakan sambungan dari satu atau proses ke proses lainnya dalam halaman atau lembar yang berbeda</p> |

Sumber : Ladjamudin (2013:263)

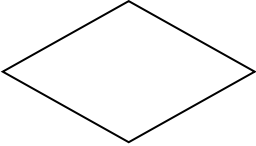

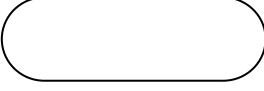
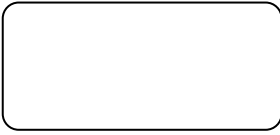
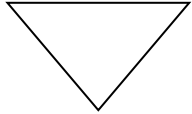

2. Processing Symbols (Simbol proses)

Simbol yang menunjukkan jenis operasi pengolahan dalam suatu proses atau prosedur. Simbol-simbol tersebut adalah sebagai berikut:

Tabel 2.5 Simbol *Processing Symbols*

| NO. | SIMBOL | KETERANGAN |
|-----|---|---|
| 1. |  | <p>Simbol <i>Offline Connector</i></p> <p>Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman atau lembar yang berbeda.</p> |
| 2. |  | <p>Simbol Manual</p> <p>Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer (manual).</p> |

Lanjutan Tabel 2.5 Simbol *Processing Symbols*



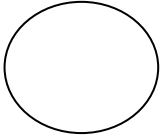



| NO. | SIMBOL | KETERANGAN |
|-----|---|---|
| 3. |  | Simbol <i>Decision</i> atau Logika Untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya atau tidak. |
| 4. |  | Simbol <i>Predefined</i> Proses Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal. |
| 5. |  | Simbol Terminal Untuk menyatakan permulaan atau akhir suatu program. |
| 6. |  | Simbol <i>Keying Operation</i> Untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i> . |
| 7. |  | Simbol <i>off-line storage</i> Untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu. |
| 8. |  | Simbol <i>Manual Input</i> Untuk memasukan data secara manual dengan menggunakan <i>online keyboard</i> . |

Sumber : Ladjamudin (2013:263)

3. Input-output Symbols (Simbol input-output)

Simbol yang menunjukkan jenis peralatan yang digunakan sebagai media *input* atau *output*. Simbol-simbol tersebut adalah sebagai berikut:

**Tabel 2.6** *Input-Output Symbols*

| NO. | SIMBOL | KETERANGAN |
|-----|---|---|
| 1 |  | Simbol <i>Input-output</i> Untuk menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya. |
| 2. |  | Simbol <i>Punched Card</i> Untuk menyatakan input berasal dari kartu atau <i>output</i> ditulis ke kartu |
| 3. |  | Simbol <i>Magnetic-tape unit</i> Untuk menyatakan input berasal dari pita <i>magnetic</i> atau output disimpan ke pita <i>magnetic</i> . |
| 4. |  | Simbol <i>Disk storage</i> Untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i> . |
| 5. |  | Simbol <i>Document</i> Untuk mencetak laporan ke printer. |
| 6. |  | Simbol <i>Display</i> Untuk menyatakan peralatan <i>output</i> yang digunakan berupa layar (video, komputer). |

Sumber : Ladjamudin (2013:263).





2.2.5 Pengertian *Blockchart*


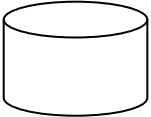

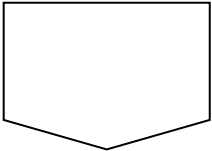
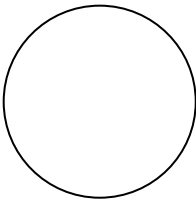
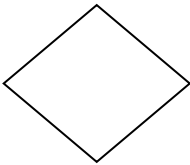
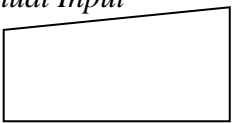
Kristanto (2008:75), *Blockchart* adalah suatu model logika data yang berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu.

Adapun simbol-simbol yang sering digunakan dalam *blockchart*, yaitu:

Tabel 2.7 Simbol-simbol *Blockchart*

| NO. | SIMBOL | KETERANGAN |
|-----|--|---|
| 1. | <p><i>Document</i></p>  | <p><i>Document</i></p> <p>Suatu simbol yang digunakan dalam menandakan suatu dokumen, bisa dalam bentuk surat, formulir, buku, benda atau berkas dan cetakan.</p> |
| 2. | <p><i>Multi Document</i></p>  | <p><i>Multi Document</i></p> <p>Suatu simbol yang digunakan dalam menandakan Multi Dokumen, bisa dalam bentuk surat, formulir, buku, benda atau berkas cetakan cetakan.</p> |
| 3. | <p><i>Manual Operational</i></p>  | <p><i>Manual Operational</i></p> <p>Suatu simbol yang digunakan dalam bentuk proses yang dilakukan secara manual.</p> |

Lanjutan Tabel 2.7 Simbol-Simbol *Blockchart*

| NO. | SIMBOL | KETERANGAN |
|-----|--|---|
| 4. | <p><i>Predefined Proses</i></p>  | <p><i>Predefined Proses</i></p> <p>Suatu simbol yang digunakan dalam proses apa saja yang tidak terdefinisi termasuk aktivitas fisik.</p> |
| 5. | <p><i>Magnetic Disc</i></p>  | <p><i>Magnetic Disc</i></p> <p>Suatu simbol yang digunakan dalam data penyimpanan (data storage).</p> |
| 6. | <p><i>Process</i></p>  | <p><i>Process</i></p> <p>Suatu proses yang dilakukan dengan menggunakan komputer.</p> |
| 7. | <p><i>Off-Page Connector</i></p>  | <p><i>Off-Page Connector</i></p> <p>Suatu simbol yang digunakan dalam termisi yang mewakili simbol-simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain.</p> |
| 8. | <p><i>Connector</i></p>  | <p><i>Connector</i></p> <p>Suatu simbol yang digunakan dalam terminasi yang mewakili simbol-simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.</p> |
| 9. | <p><i>Decision</i></p>  | <p><i>Decision</i></p> <p>Suatu simbol yang digunakan dalam pengambilan keputusan.</p> |
| 10. | <p><i>Manual Input</i></p>  | <p><i>Manual Input</i></p> <p>Suatu simbol yang digunakan dalam pemasukan data secara manual.</p> |

Sumber : Kristanto (2008:75).



2.3 Teori Pemrograman

2.3.1 PHP (*Hypertext Preprocessor*)

2.3.1.1 Pengertian PHP

Winarno (2014:49), *php* (*Hypertext Preprocessor*) adalah sebuah bahasa pemrograman *web* berbasis *server* (*server-side*) yang mampu memarsing kode *PHP* dari kode *web* dengan ekstensi *php*, sehingga menghasilkan tampilan *website* yang dinamis di sisi *client* (*browser*).

Jadi, *PHP* adalah bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis *web*.

2.3.1.2 Memasukkan Kode PHP

Winarno (2014:64), ada empat jenis tag yang bisa digunakan untuk memasukkan kode *PHP*, yaitu:

Tabel 2.8 Tag Memasukkan Kode *PHP*

| No. | Jenis Tag | Tag Pembuka | Tag Penutup |
|-----|-------------------|----------------------------|-------------|
| 1. | Tag standar | <?php | ?> |
| 2. | Tag pendek | <? | ?> |
| 3. | Tag ASP | <% | %> |
| 4. | Tag <i>script</i> | <script language="php"> | </script> |

(Sumber: Winarno, 2014:63)

Di dalam tag kode *PHP* juga terdapat komentar, ada 3 cara untuk membuat komentar di dalam tag *PHP* yaitu:

1. Tag `//` digunakan untuk komentar dalam satu baris,
2. Tag `#` juga digunakan untuk komentar dalam satu baris, dan
3. Tag `/*.....*/` digunakan apabila jumlah keterangan lebih dari 1 baris.

Nugroho (2013:157), *PHP* memiliki perintah ***echo*** dan ***print*** untuk menampilkan apapun ke layar. Lebih baik gunakan perintah ***echo*** pada semua program yang dibuat supaya program bisa dijalankan pada semua paket aplikasi *PHP*. Dalam menggunakan perintah ***echo*** setiap objek yang berbentuk teks harus kita kait dengan **petik ganda** (“”), sedangkan untuk objek yang berbentuk angka bisa diabaikan. Setiap satu baris perintah diakhiri dengan tanda **titik koma** (;), itu



wajib jadi selama belum ada tanda **titik koma**, maka pembacaan kode akan dilanjutkan pada baris berikutnya.

2.3.1.3 Variabel *PHP*

Winarno (2014:64), Variabel merupakan sebuah istilah yang menyatakan sebuah tempat yang digunakan untuk menampung nilai-nilai tertentu dimana nilai di dalamnya bisa diubah-ubah. Di *PHP* variabel dikenal dengan adanya tanda (\$) yang kemudian bisa diikuti dengan angka, huruf, dan *underscore*. Namun, variabel tidak bisa mengandung spasi.

2.3.2 *Responsive Web Design*

Riyanto (2014:12), *Responsive Web Design* adalah *client-side* yang berarti halaman akan dikirim ke *browser* perangkat dan *browser* kemudian memodifikasi tampilan halaman yang berkaitan dengan ukuran jendela *browser*.

2.3.3 CSS (*Cascading Style Sheet*)

2.3.3.1 Pengertian CSS (*Cascading Style Sheet*)

Koesheryatin (2014:101), *Cascading Style Sheet* adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu *website*, baik tata letaknya, jenis huruf, warna, dan semua yang berhubungan dengan tampilan. Pada umumnya CSS digunakan untuk memformat halaman *web* yang ditulis dengan HTML (*Hypertext Markup Language*).

2.3.3.2 Cara Menuliskan CSS

Koesheryatin (2014:101), ada dua cara yang bisa diterapkan untuk menggunakan CSS langsung di dalam satu file HTML (*internal/inline style sheet*). Cara yang kedua dengan memanggil CSS tersebut dari file CSS tersendiri (*external style sheet*).

2.3.4 Pengertian JS (*Java Script*)

Winarno (2014:129), *Java Script* adalah bahasa *scripting* yang banyak digunakan oleh programer *web* untuk memberikan efek pemrograman di halaman.



2.3.5 Pengertian XAMPP

Nugroho (2013:1), xampp adalah perangkat lunak (*free software*) bebas, yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Fungsi *XAMPP* sendiri adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain : *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP*.

2.3.6 Pengertian Basis Data (*Database*)

Kadir (2010:2) menjelaskan, *Database* (basis data) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Dalam hal ini, pengertian akses dapat mencakup pemerolehan data maupun manipulasi data, seperti menambah dan menghapus data.

Indrajani (2015:70), Basis Data adalah sebuah kumpulan data yang saling berhubungan secara logis, dan merupakan sebuah penjelasan dari data tersebut, yang didesain untuk menemukan data yang dibutuhkan oleh sebuah organisasi.

Istilah-istilah yang berhubungan dengan database sebagai berikut:

1. *Character* merupakan bagian data terkecil, dapat berupa karakter numerik, huruf ataupun karakter-karakter khusus (*special characters*) yang membentuk suatu item data atau *field*.
2. *Field* adalah suatu atribut dari *record* yang menunjukkan suatu item dari data, seperti misalnya nama, alamat, dan lain sebagainya. Kumpulan dari *field* membentuk suatu *record*.
3. *Record* adalah kumpulan dari *field* membentuk suatu *record*. *Record* menggambarkan suatu unit data individu yang tertentu. Kumpulan dari *record* membentuk suatu *file*. Misalnya *file* personalia, tiap *record* dapat mewakili data tiap karyawan.
4. *File* terdiri dari *record-record* yang menggambarkan satu kesatuan data yang sejenis



Jadi, basis data adalah suatu data yang diletakkan di dalam sebuah tempat sehingga memudahkan kita untuk mengaksesnya.

2.3.7 *phpMyAdmin*

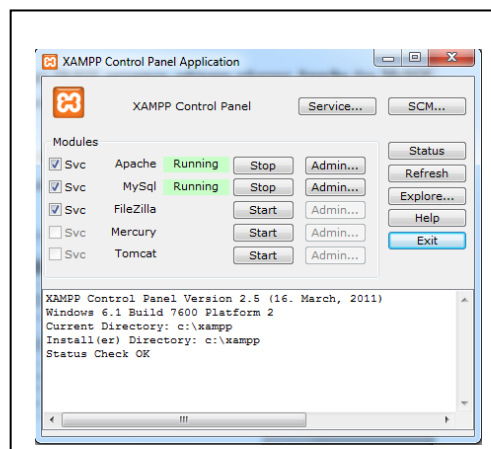
2.3.7.1 Pengertian *phpMyAdmin*

Nugroho (2013:71), *phpmyadmin* adalah *tools MySQL Client* berlisensi *freeware*, anda dapat mengunduhnya secara gratis pada situs resminya, *phpmyadmin* harus dijalankan di sisi *server web* dan pada komputer harus tersedia *PHP*, karena berbasis *web*.

2.3.7.2 Menjalankan *phpMyAdmin*

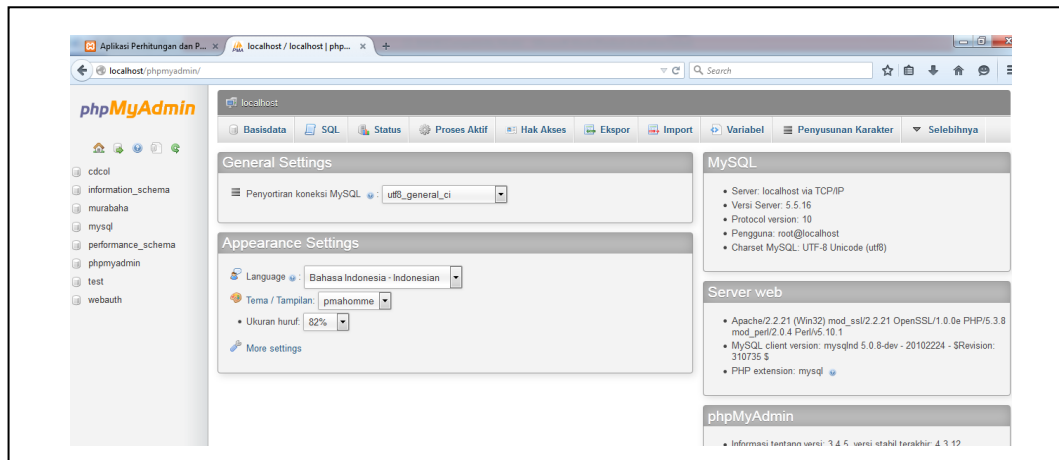
Nugroho (2013:72), Cara menjalankan *phpmyadmin* adalah sebagai berikut:

1. Buka **XAMPP Control Panel** lalu jalankan modul servis **Apacheweb server**.
2. Jalankan juga **MySQL servernya**, sehingga sekarang **Apache** dan **MySQL** statusnya *running*.



Gambar 2.1 XAMPP Control Panel

3. Sekarang buka *web browser*, misalnya **Internet Explore** atau **Firefox**
4. Pada kotak alamat, ketikkan; <http://localhost/phpmyadmin/> tekan enter.



Gambar 2.2 phpmyadmin

2.3.8 MySQL

2.3.8.1 Pengertian MySQL

Kadir (2013:15), *MySQL* adalah nama *database server*. *database server* adalah *server* yang berfungsi untuk menangani *database*. *Database* adalah suatu pengorganisasian data dengan tujuan memudahkan penyimpanan dan pengaksesan data. Dengan menggunakan *MySQL*, kita bisa menyimpan data dan kemudian data bisa diakses dengan cara yang mudah dan cepat. *MySQL* tergolong sebagai *database* relasional. Pada model ini, data dinyatakan dalam bentuk dua dimensi yang secara khusus dinamakan tabel. Tabel tersusun atas baris dan kolom.

2.3.8.2 Fungsi-Fungsi MySQL

eWolf Community (2014:206), fungsi yang berawalan **mysql_** yang digunakan untuk mengakses *database server MySQL* yang dibahas di bawah ini:

Tabel 2.9 Fungsi-Fungsi MySQL

| No. | Nama Fungsi | Kegunaan |
|-----|-----------------|--|
| 1. | mysql_connect() | Digunakan untuk menghubungkan dokumen dengan <i>database MySQL</i> |
| 2. | mysql_close() | Digunakan untuk memutus hubungan dengan <i>database MySQL</i> |

**Lanjutan Tabel 2.9** Fungsi-Fungsi *MySQL*

| No. | Nama Fungsi | Kegunaan |
|-----|--|---|
| 3. | <code>mysql_affected_rows()</code> | Digunakan untuk mendapatkan jumlah baris (record) yang dihasilkan dari operasi MySQL sebelumnya |
| 4. | <code>mysql_fetch_array()</code> <code>mysql_fetch_assoc()</code> | Digunakan untuk mendapatkan array dari suatu recordset |
| 5. | <code>mysql_fetch_field()</code> | Digunakan untuk mendapatkan sebuah objek yang mengandung informasi field dari sebuah recordset |
| 6. | <code>mysql_fetch_lengths()</code> | Digunakan untuk mendapatkan panjang baris pada setiap isi field |
| 7. | <code>mysql_fetch_object()</code> | Digunakan untuk menghasilkan baris dari recordset sebagai sebuah objek |
| 8. | <code>mysql_fetch_row()</code> | Digunakan untuk menghasilkan array numerik dari baris pada sebuah recordset |
| 9. | <code>mysql_field_len()</code> | Digunakan untuk mendapatkan informasi panjang maksimum field dalam sebuah recordset |
| 10. | <code>mysql_field_seek()</code> | Digunakan untuk melompat ke field tertentu dalam sebuah recordset |
| 11. | <code>mysql_get_client_info()</code> <code>mysql_get_host_info()</code> | Digunakan untuk mendapatkan informasi tentang MySQL client |
| 12. | <code>mysql_get_server_info()</code> | Digunakan untuk mendapatkan informasi tentang MySQL server |
| 13. | <code>mysql_info()</code> | Digunakan untuk mendapatkan informasi tentang <i>query</i> terakhir |

**Lanjutan Tabel 2.9** Fungsi-Fungsi *MySQL*

| | | |
|-----|---------------------------------|--|
| 14. | <code>mysql_num_fields()</code> | Digunakan untuk mendapatkan informasi jumlah <i>field</i> pada suatu <i>recordset</i> |
| 15. | <code>mysql_num_rows()</code> | Digunakan untuk mendapatkan informasi jumlah baris pada <i>recordset</i> |
| 16. | <code>mysql_ping()</code> | Digunakan untuk memeriksa koneksi <i>server</i> dan akan mencoba untuk melakukan koneksi ulang jika koneksi terputus |
| 17. | <code>mysql_query()</code> | Digunakan untuk menjalankan <i>query</i> pada <i>databaseMySQL</i> |
| 18. | <code>mysql_result()</code> | Digunakan untuk mendapatkan nilai dari <i>field</i> dalam sebuah <i>recordset</i> |
| 19. | <code>mysql_select_db()</code> | Digunakan untuk menentukan (set) <i>database MySQL</i> yang aktif |
| 20. | <code>mysql_stat()</code> | Digunakan untuk menghasilkan informasi status <i>server</i> terkini |

(Sumber: eWolf Community, 2014:206)

2.3.9 Adobe Dreamweaver CS3

2.3.9.1 Pengertian Adobe Dreamweaver CS3

Madcoms (2008:1), *dreamweaver* adalah sebuah *HTML (Hyper Text Markup Language)* editor profesional untuk mendesain *web* secara visual dan mengelola situs atau halaman *web*.

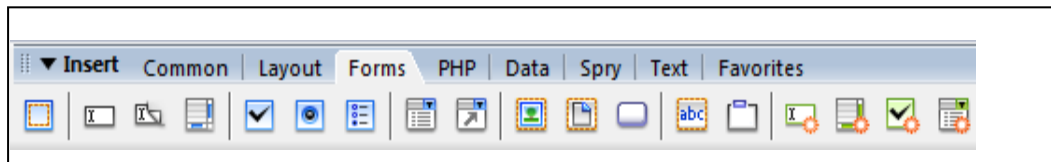
2.3.9.2 Ruang Kerja Adobe Dreamweaver CS3

Madcoms (2008:2), ruang kerja dasar *Dreamweaver CS3* adalah sebagai berikut:



1. *Insert Bar*

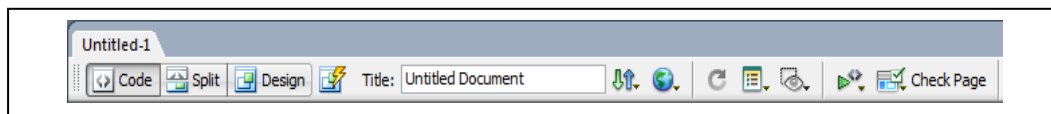
Berisi tombol-tombol untuk menyisipkan berbagai macam objek seperti: *image*, tabel dan *layer* ke dalam dokumen. *Insert Bar* terdiri dari beberapa kategori sesuai dengan jenis yang disisipkan.



Gambar 2.3 Tampilan *Insert Bar*

2. *Document Toolbar*

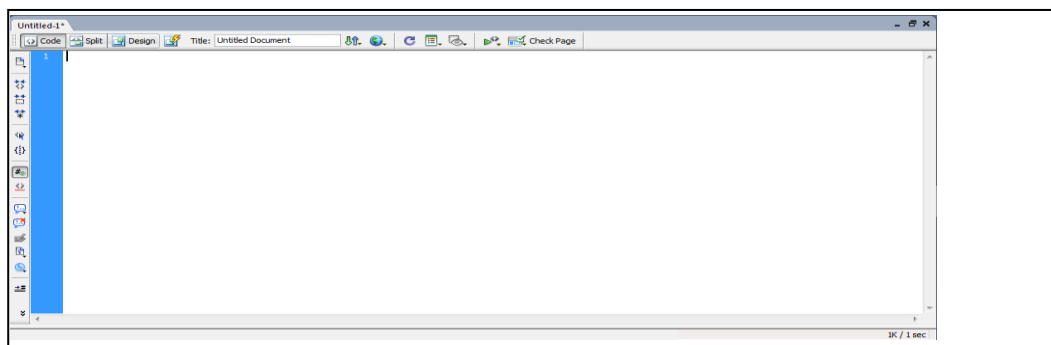
Berisi tombol-tombol dan menu *pop-up* yang menyediakan tampilan berbeda dari jendela dokumen. Anda bisa mengatur tampilan jendela dokumen dalam bentuk *design view* atau *code view* atau kedua-duanya. Anda juga dapat memberikan judul jendela dokumen pada bagian *title*.



Gambar 2.4 Tampilan *Document Toolbar*

3. *Coding Window*

Berisi kode-kode HTML dan tempat untuk menuliskan kode-kode pemrograman, misalnya *PHP* atau *ASP*.



Gambar 2.5 Tampilan *Coding Window*

4. *Panel Group*

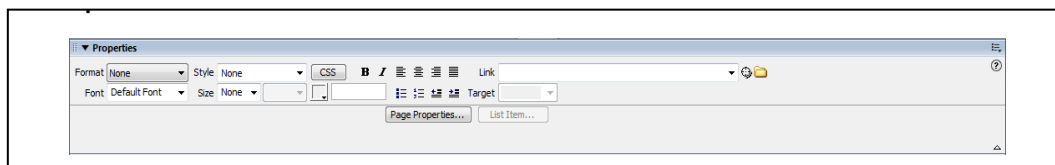
Adalah kumpulan *panel* yang saling berkaitan satu sama lainnya yang dikelompokkan di bawah satu judul.



Gambar 2.6 Tampilan *Panel Group*

5. *Property Inspector*

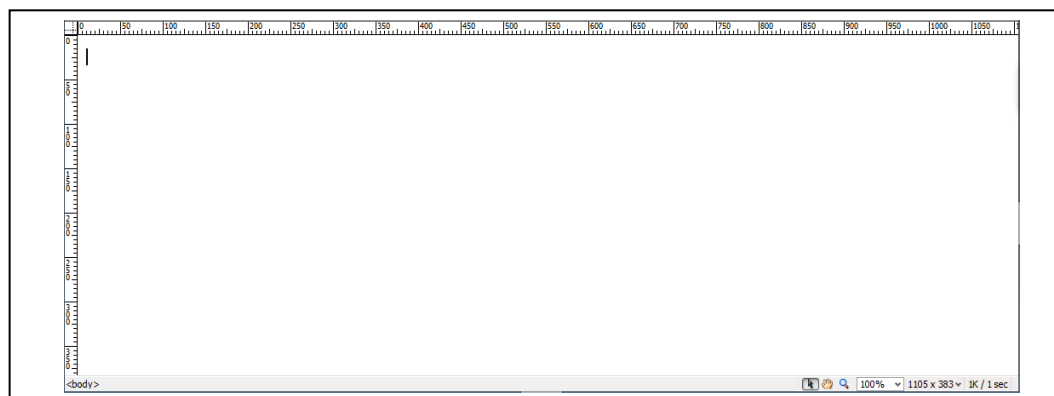
Digunakan untuk melihat dan mengubah berbagai properti objek atau teks



Gambar 2.7 Tampilan *Property Inspector*

6. Jendela Dokumen

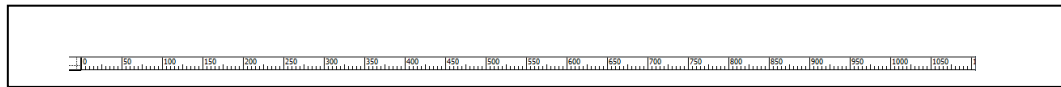
Berfungsi untuk menampilkan dokumen di mana Anda sekarang bekerja.



Gambar 2.8 Tampilan Jendela Dokumen

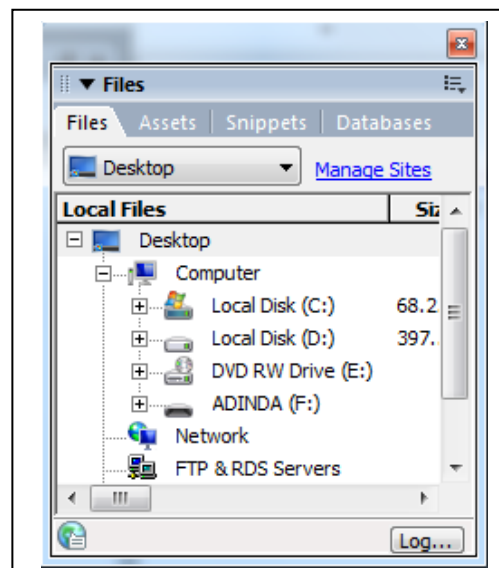
7. *Ruler*

Digunakan untuk memudahkan pengukuran saat Anda mendesain suatu halaman *web*.

**Gambar 2.9** Tampilan *Ruler*

8. *Site Panel*

Digunakan untuk mengatur *file-file* dan *folder-folder* yang membentuk situs *web* Anda.

**Gambar 2.10** Tampilan *Site Panel*